



# オフショア開発を成功させる SEPG活動

ソニーデジタルネットワークアプリケーションズ(株)  
杉村 宗泰 (nsugimura@sdna.sony.co.jp)

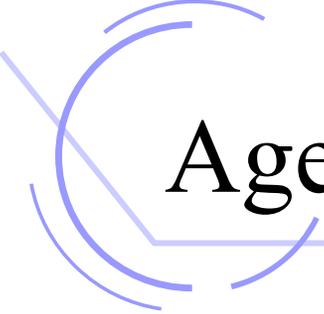


# 会社概要

---

## 4 Sony Digital Network Applications Inc.

- 設立 2000年8月
- 資本金 4億5000万円  
ソニー(株)100%出資
- 従業員数 256名(2006年3月現在)
- 年商 50億円 (2005年度実績)
- URL <http://www.sonydna.com>



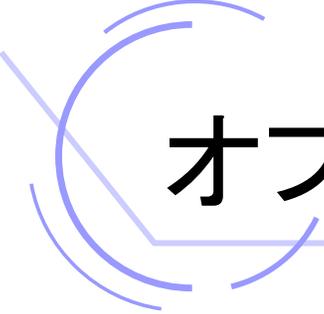
# Agenda

---

- ㊦ 発表のポイント
- ㊦ オフショア開発のきっかけ
- ㊦ 弊社のオフショアの歴史
  - 第1期 - 立ち上げの時代
  - 第2期 - 問題解決の時代
  - 第3期 - 改善の時代（現在）
  - 第4期 - 充実の時代（今後）
- ㊦ まとめ

# 発表のポイント

- ④ **商品化、製品開発視点のプロセス構築**
  - **組織都合重視**のプロセス構築は破綻しやすい
    - ↳ 組織の都合＝文化の違いが顕著に現れる部分
- ④ **プロセスを明確化することの重要性**
  - 開発活動の**ブラックボックス化**はリスクの塊
  - プロセス明確化は**空洞化防止の切り札**
- ④ **データに基づいた改善活動**
  - **お互いが合意したデータ**に基づいて改善活動を行う



# オフショア開発のきっかけ

---

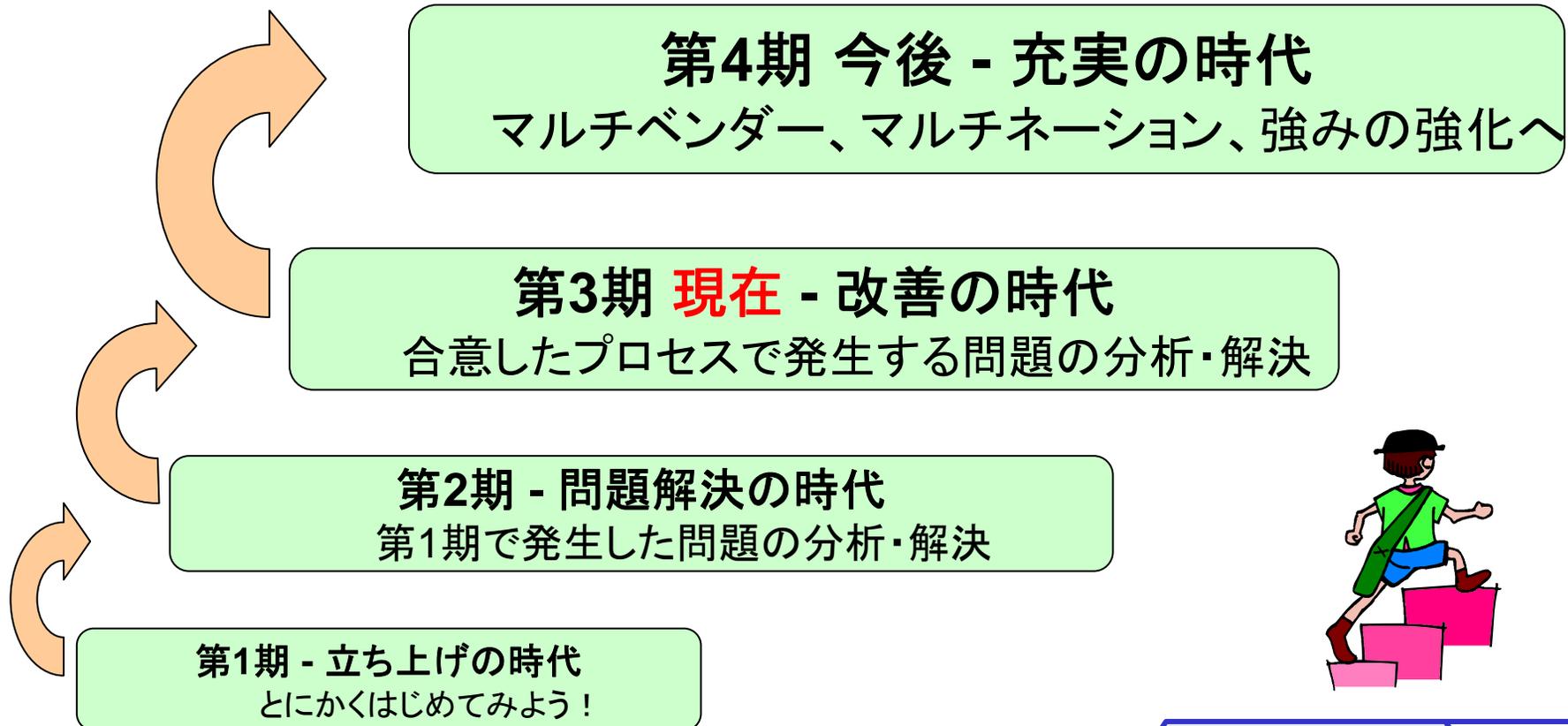
ヒューマンリソースが足りない

集中すべき領域に要員をシフトしたい

コスト競争力を向上させたい

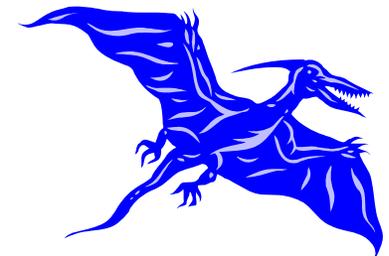
分散開発体制を構築したい

# オフショア開発の歴史



# 第1期 - 立ち上げの時代

- 4 幾つかのPJをトライアル的に発注
  - o 色々な問題が出ていたが、特に解決策はなし
  - o とにもかくにも発注は継続……
- 4 この時点ではSEPGの関与はなし
  - o 現場のPL主導での立ち上げ
    - u 多忙のため会社間でのプロセス合意までは手が回らず
- 4 トライアルを終え、本格的に発注を開始
  - o 通信設備や開発設備などのインフラは整備されていたが、、
    - u 開発プロセス、品質に関する合意はなかった



第1期 - 立ち上げの時代

# オフショア開発を本格的に開始-1

- 4 やっぱり？多くの問題が発生
  - ソフトウェアエンジニアリング的な問題
  - プロジェクトマネジメント的な問題
  - 文化の違いや言葉の問題
- 4 このままいくと・・・やっぱり
  - 中国は品質が悪い
  - オフショア先に**マネジメント能力は期待できない**
  - 自分達でやったほうが早い
  - **コア技術の流出・空洞化**につながるのではないか？
  - 中には**自分の仕事がなくなる**んじゃないか・・・の声



# オフショア開発を本格的に開始-2

- 4 個人的に見てきたアンチパターンが脳裏をよぎる
  - o 無策なPLの無謀な発注経験が全体の雰囲気となり尻すぼみ
    - u 私の考えていることが出来ていない
    - u 行間を読んでくれない
    - u 言ったこと以上の事はしてくれない



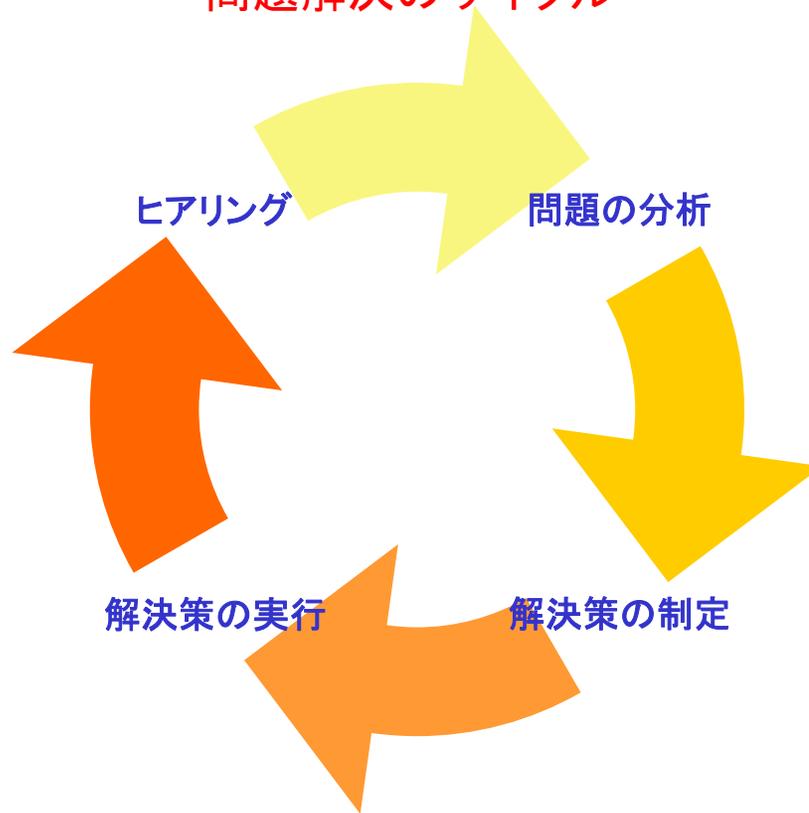
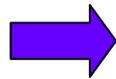
- o 会社目標としてオフショアを推進
  - u XX部門は上期YY個のプロジェクトを発注しなさい
  - u XX円以上の予算のプロジェクトはオフショアを必須
  - u 多くのオフショアを出した部門を表彰



# 第2期 - 問題解決の時代

## 問題解決のサイクル

SEPGがプロセス面の  
整備・問題解決に着手

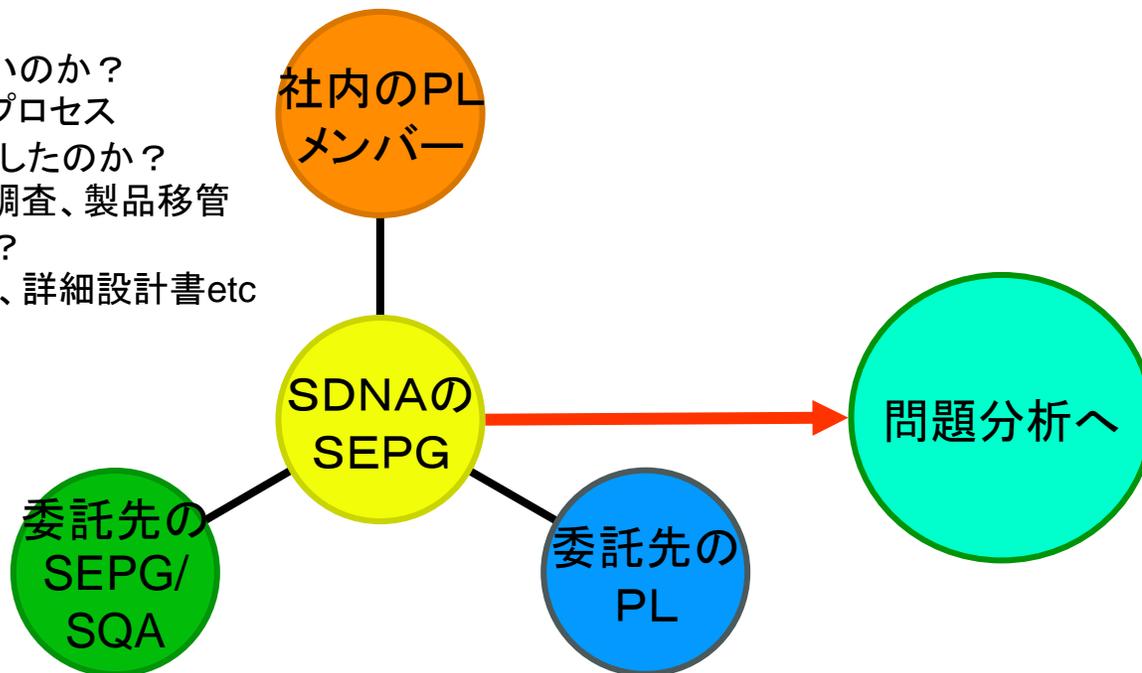


改善要望

## 第2期 - 問題解決の時代

# ヒアリング

- どのような計画で発注をしたいのか？  
委託期間、工数、開発プロセス
- どのような特性の開発を依頼したのか？  
新規開発、派生開発、調査、製品移管
- どのような資料を渡したのか？  
機能一覧、要求仕様書、詳細設計書etc



- 出された計画をどのように理解したのか？
- 内部計画はどのように立てたのか？
- 進捗管理はどのように行ったのか？

- どのような開発プロセスを適用したのか？
- 要求された製品に合ったプロセスであったのか？
- プロセス上の問題をどのように解決したのか？

## 第2期 - 問題解決の時代

# 出された問題の分析

## 問題の原因は限定的

### 依頼する開発の特性に対してお互い合意したベースプロセスが存在していない

- PLの今までイメージ(経験)で発注を行っていた
- プロセス不備は技術流出の温床となっていた



- 委託元としては、先方の開発がブラックボックス化
- 委託先にとっては、毎回異なる開発方法での対応を迫られた
- 丸投げしか選択肢が無い

### 進捗を正しく把握する方法が合意されていなかった

- XX%終了したとなっているが、その根拠は？
- 相変わらずの「言ったはず」「お願いしたはず」



- 作業の終了状態の合意が行なわれていない
- コミュニケーションに情報が伴っていない

### 見積りの合意が上手に出来なかった

- XX人月が多い、少ないという不毛な議論
- 見積りプロセスが異なるため調整出来ない



- 数値の根拠の合意が行なわれていない

### 品質が悪い・生産性が悪い(自分でやったほうがいい)

- 客観的な判断が行えていない
- 合理的な問題解決が行われていない



- データに基づいていない
- 問題解決をどちらかに押し付ける

第2期 - 問題解決の時代

# 解決策の制定

依頼する開発の特性に対してお互い合意したベースプロセスが存在していない

開発特性に応じたベースプロセスの構築・合意

見積もりの合意が上手に出来なかった

サイズ見積もりをベースに見積りを行うことの合意

進捗を正しく把握する方法が合意されていなかった

簡易版EVMの導入

品質が悪い・生産性が悪い(自分でやったほうがいい)

品質・生産性に関する計測プロセスを整備

## 第2期 - 問題解決の時代

# 開発特性に応じたベースプロセスの構築・合意

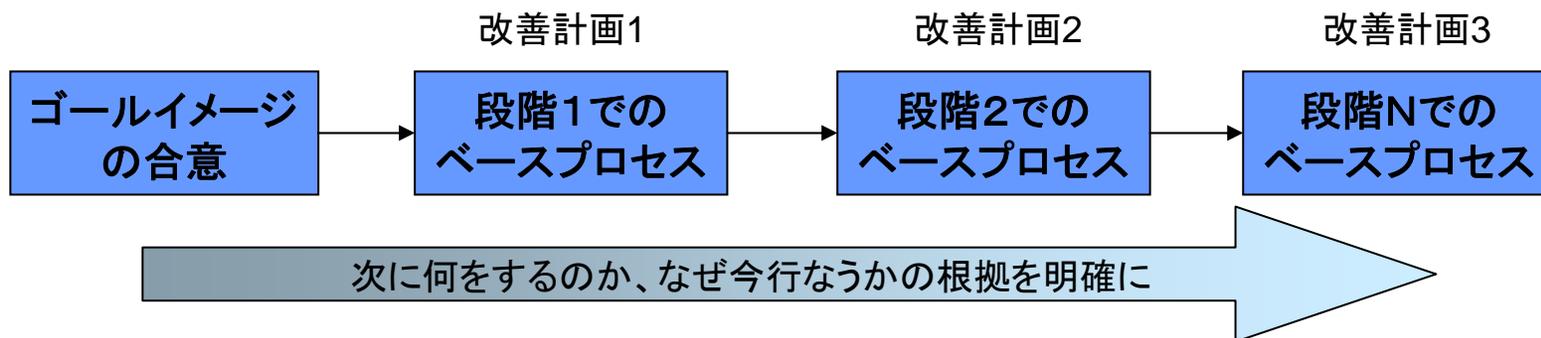
- ④ 自社の製品開発の特性から、ベースプロセスを定義
  - 新規開発プロセス
  - 派生開発プロセス
  - 検証プロセス
  - インストーラー作成プロセス
  - パッチ作成プロセス
- ④ ベースプロセスを可視化
  - **PFD** (Process Flow Diagram)でプロセスを表現
    - ↳ [http://homepage3.nifty.com/koha\\_hp/process/PFDform2.pdf](http://homepage3.nifty.com/koha_hp/process/PFDform2.pdf)
- ④ ベースプロセスの合意
  - 委託先の、PM、SEPG、SQAと開発プロセスについて合意
- ④ 改善計画見通しの合意



## 第2期 - 問題解決の時代

# ベースプロセス制定時の注意点

- 4 製品開発全体からプロセスを構築する
  - 自社では、の視点からプロセスを構築する溝を作るだけ
  - **お互いの都合**の優先度を上げると、問題解決は出来ない
- 4 全体から見た責任範囲を定義する
  - 製品開発の視点での役割分担、作業分担を定義する
    - ↳ お互いの都合ではなく、**製品開発上の都合**に重心を置く
- 4 現場主義、理想主義を捨てる
  - どちらに偏ってもプロセス制定は失敗する
- 4 段階的な改善計画見通しを同時に制定する



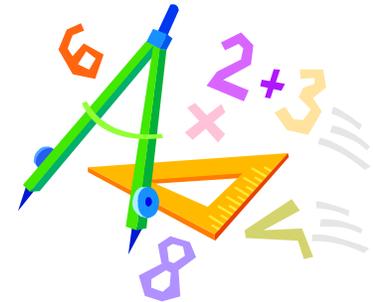
## 第2期 - 問題解決の時代

# サイズ見積もりをベースに見積りを行うことの合意・履行

- 4 PFDから開発に必要な**成果物を洗い出し**
  - 何を作るのか？が工数見積りの基本
    - ↳ XXを行なう、という表現からは工数は導き出せない
- 4 成果物毎に**予想されるサイズ(量)を見積もる**
  - クラス設計 → クラスはXX個
  - ソースコード → XX行
- 4 予想したサイズと生産性実績から、必要な工数を見積もる
  - 要求仕様書
    - ↳ 工数 = 仕様XX個 ÷ 仕様を1つ定義するのに必要な時間
- 4 成果物毎の**サイズ見積もりをお互いが行ない**、差異の出る部分について調整を行なう
  - XX機能を実現するにはYY人月の工数が必要

↓

  - XX機能を実現するには、
    - ↳ 新規開発プロセスを適用した場合、必要な成果物はA、B、Cである
    - ↳ 成果物AはサイズをYYと見込んでおり、標準的な生産性から予想される工数はZZ人月
  - **差異の出る部分が明確**になり、見積もり値を合理的に判断し調整することが可能になる



## 第2期 - 問題解決の時代

# 簡易版EVMの導入

- 4 見積もりで導出した**サイズを基に進捗管理**
  - **週報**のフォーマットをEVM的に改良
    - ↳ 予定期間
    - ↳ 予定サイズ
    - ↳ 実績期間
    - ↳ 実績サイズ
- 4 **マイルストーン定義の合意**
  - 設計完了、コーディング完了とは？
    - ↳ **どのような状態**を完了と定義するのかを議論、合意
    - ↳ 例えば、コーディング完了の状態とは、
      - ↳ 正常系、異常系を含めコーディングが完了し、以降デバッグ以外にはソースコードを修正する必要の状態
      - ↳ コンパイルが完全な形で通る状態
      - ↳ ソースコードレビューが完了し、指摘事項が反映されている状態等
  - **ISO時代の定義**を有意義に活用



## 第2期 - 問題解決の時代

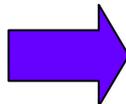
# 品質・生産性に関する計測プロセスを整備

- ④ 計測指標、計測データの合意
  - 何をどのような目的で、どのように計測するのか
  - 得られた指標からどのようなアクションを行なうのか
- ④ 計測データは週ベースで管理する
  - 進捗報告(EVM)にそった形で計測プロセスを整備



工程・工数計測

工程	工数(h)
調査	100h
設計	150h
製造	100h
テスト	100h



工程・工数/成果物・サイズ計測

工程	成果物	工数(h)	サイズ
調査	調査報告	100h/120h	50項目/50項目
設計	概念クラス図	200h/250h	150クラス/200クラス
	シーケンス図	50h/40h	30パターン/40パターン
製造	ソースコード	250h/350h	5000行/20000行
テスト	テストケース	50h/50h	200個/200個
	テスト結果	0h/100h	0個/200個

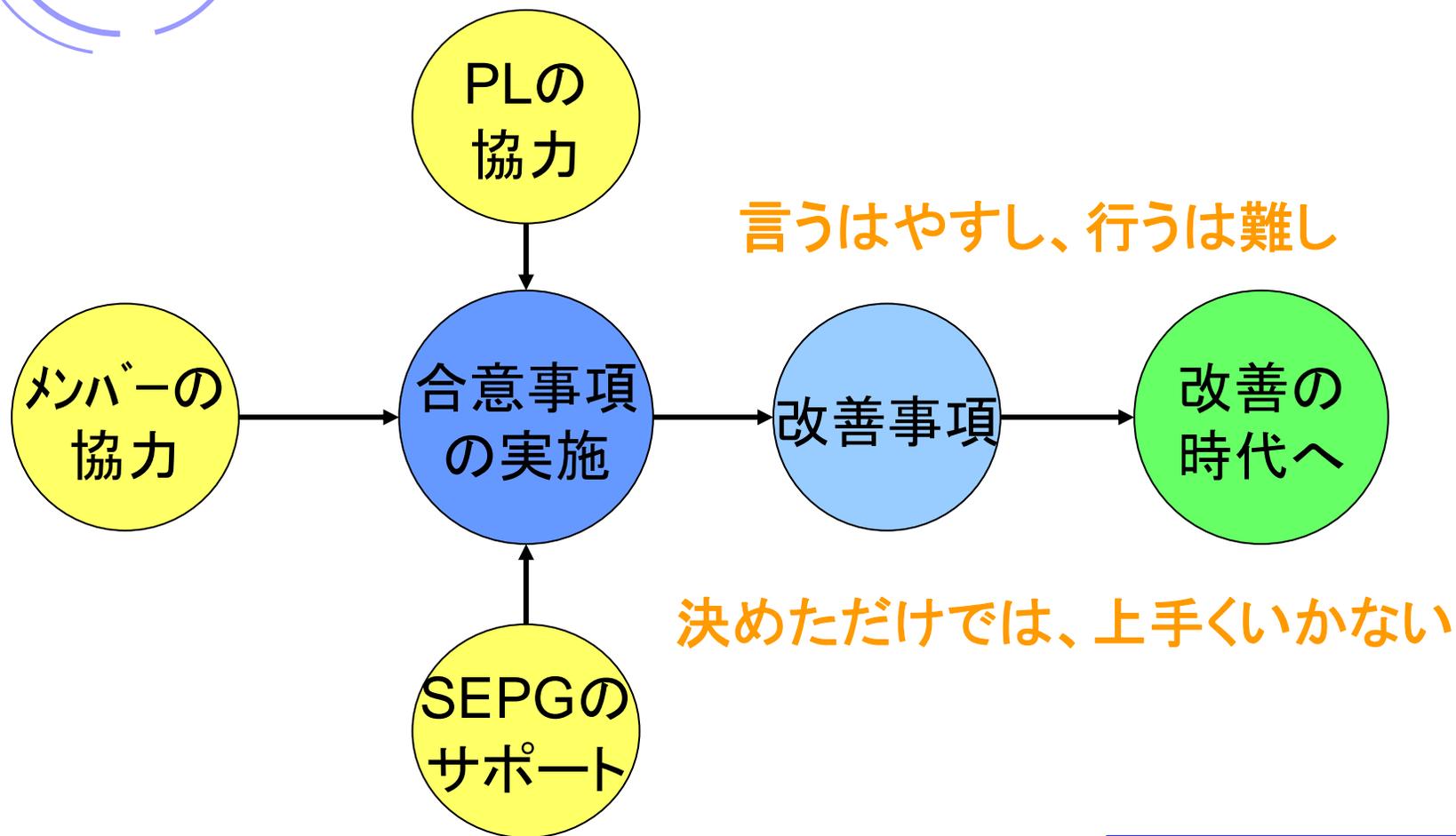
## 第2期 - 問題解決の時代

# 計測指標 ～ 参考

- 4 ソースコードの生産性
  - コーディングプロセスでの工数あたりのソースコードの生産性(LOC/時/プロセス)
- 4 スケジュールあたりのソースコードの生産性
  - 実装フェーズでのスケジュールあたりのソースコード生産性(LOC/日/フェーズ)
- 4 プロセス毎の成果物生産性
  - 各フェーズで作成される成果物の生産性(指標はさまざま)
- 4 全工程あたりのソースコード生産性
  - 開発活動全ての工数・スケジュールあたりのソースコード生産性(LOC/1プロジェクト)
- 4 ソースあたりの欠陥数
  - ソースコード行あたり何個の欠陥があるか(バグ数/KLOC)
- 4 プロセスあたりの欠陥数
  - プロセスでの工数あたりのバグ数(バグ数/プロセス工数)
    - ↳ どのプロセスで作りこまれた欠陥か? どのプロセスで発見された欠陥か?
- 4 レビュー効率
  - レビュー単位時間でどれだけ問題を指摘することが出来たか?

第2期 - 問題解決の時代

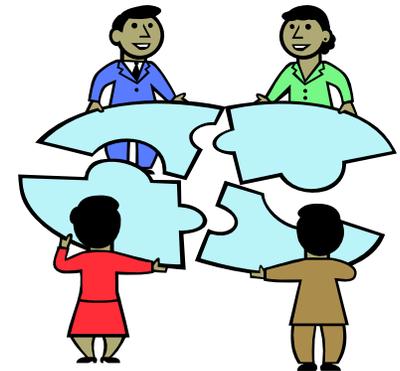
# 解決策の実行



# 第3期 現在 - 改善の時代

## 合意したプロセスで発生する問題の分析・解決

- 4 SEPG/SQAが定期的にプロセス/品質に関して情報交換
  - 修正の必要なプロセスについての合意
  - 問題の起こっているプロジェクトについての支援
  - 経営からの要望の反映
- 4 定期的なプロセスの見直し
  - 計測データ・現場からの要望に応じてプロセスを修正
    - ↳ QCDの安定した開発活動の継続
  - 市場の要求に応えられる開発形態への対応
    - ↳ 最適な商品化プロセスの提案、定義
- 4 信頼感のさらなる向上
  - お互い様の精神が重要
    - ↳ 問題の原因はハッキリさせるが、問題解決には双方で取り組む



# 第4期 今後 - 充実の時代

マルチベンダー、マルチネーション、強みの強化へ

## 4 リスク回避

- 諸国の事情に起因するリスク
- 企業経営に関するリスク



## 4 空洞化回避

- 筋肉を強化し、贅肉を落とすアウトソース本来の姿へ
  - ↳ 工程分割、機能分割だけでは実現出来ない
  - ↳ 新たなルール・文化の構築へ
    - ≡ 物作りの本質が蓄えられる開発戦略

## 4 相互尊重

- 目的達成のためのパートナーとして、お互いを更に尊重



# まとめ

- ④ プロセス構築が成功への第一歩
  - 新しい開発プロセスの共有・合意・履行
- ④ 開発の共通言語はプロセス
  - エンジニアリング技術に基づいた開発プロセスは、文化・習慣・言葉を超える
- ④ 数値目標の達成方法を共有する
  - 「日本のやり方」では絶対に通じない